Chapitre ①: Systèmes De Numération.

I- Signaux analogiques et numériques:

1- Signal analogique:

Un signal analogique c'est un signal où on peut avoir une infinité de valeur.

• Signal numérique:

c'est un signal où le nombre de valeurs est dénombrable.

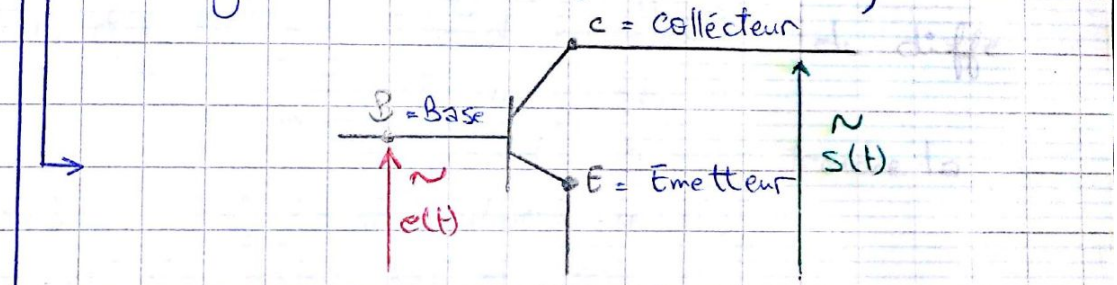
• Signal binaire: 2 valeurs de signal.

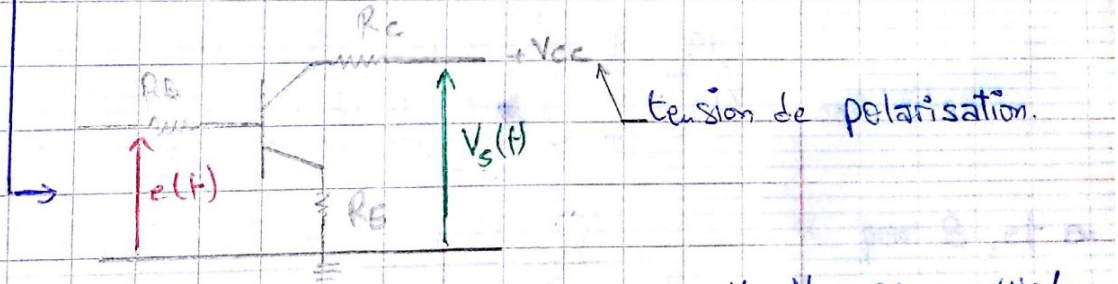
Exemple 1: VRAI / FAUX ; OUVERT / FERME.

Exemple 2: un transistor bipolaire.

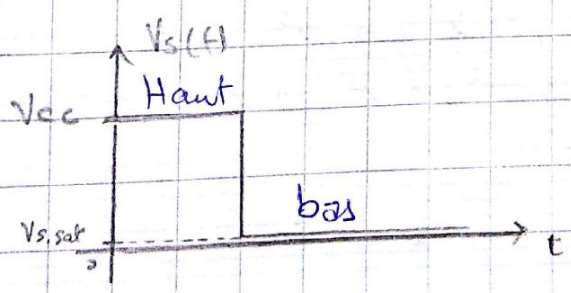
→ transistor en régime linéaire: $S(t) = A_e(t)$

→ en régime non linéaire (de commutation)





- transistor en commutation: $V_s(t)$
  - $V_s = V_{cc} = 5V \rightarrow e(t) \not> 0$
  - $V_s = V_{s,sat} = 0,2V \rightarrow e(t) > 0$



Scanned by CamScanner

La sortie est soit à l'état Haut (+Vcc) ; soit à l'état bas (0V)

→ Le transistor inverseur ⇒ Système binaire.

• Codification :

état H → "1" ; état bas → "0"

→ en décimal = 10 états (0;1;2;...;9) base = 10

→ en binaire = 2 états (0;1) base = 2.

2 - Systèmes de numérations :

2.1 - Définition :

La fonction des systèmes de numération est de représenter les valeurs numériques le plus simplement et le plus clairement possible.

2.2 - Differents systèmes de numération :

X : un nombre qcq : peut être représenter suivant différents systèmes de numération.

Pour une base B qcq : $X_B = b_{n-1}B^{n-1} + b_{n-2}B^{n-2} + \dots + b_0B^0 + b_{-1}B^{-1} + b_{-m}B^{-m}$.

b_i = poids de rang i.

B = base

n = nombre de chiffres $0 \leq b_i \leq B-1$

ⓐ - la base décimale B = 10.

$B_{10} = \{0;1;2;\dots;9\}$

$19_{10} = 1.10^1 + 9.10^0 = 19$

$(19,25)_{10} = \underbrace{1.10^1 + 9.10^0}_{\text{entière}} + \underbrace{2.10^{-1} + 5.10^{-2}}_{\text{Fractionnaire}}$

ⓑ - la base binaire B = 2

$B_2 = \{0;1\}$

$19_{10} = 16 + 2 + 1$

$= 1.2^4 + 0.2^3 + 1.2^1 + 1.2^0 = 10011)_2$

$111,101)_2 = 1.2^2 + 1.2^1 + 1.2^0 + 0.2^{-2} + 1.2^{-1} = 7,625)_{10}$

Scanned by CamScanner

ⓒ la base octale B = 8 :

$B_8 = \{0; 1; 2; \ldots; 7\}$

$144)_8 = 1.8^2 + 4.8^1 + 4.8^0 = 64 + 32 + 4 = 100)_{10}$

ⓓ Base Duodécimale B = 12 :

$B_{12} = \{0; 1; 2; \ldots; 9; A; B\}$

ⓔ Base héxadécimale B = 16 :

$B_{16} = \{0; 1; 2; \ldots; 9; A; B; C; D; E; F\}$.

$1EF)_{16} = 1.16^2 + E.16^1 + F.16^0 = 256 + 16 \times 14 + 15 = 495$.

2.3 Passage d'une base à une autre : Transcodage.

ⓐ Passage d'une base qcq à B_{10} :

Règles de Passage : la valeur décimale s'obtient par sa forme polinomiale avec les règles de calcul habituelles dans le syst décimale (voir TD).

ⓑ Conversion d'un nombre décimal x en un nombre de base qcq :

Méthode ① : • Dresser une table du difference puissances B à la puissance B^i.

• à x donné, on retronche la plus grande puissance de B possible.

Exemple : $X = 262)_{10} \longrightarrow B = 2$.

$2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8 = 256$.

$262 - 256 = 6 \longrightarrow 1.2^8$.

$6 - 2^2 = 2 \longrightarrow 1.2^2$ $\Big) \longrightarrow 100000110$

$2.2^1 = 0 \longrightarrow 1.2^1$

méthode ② : On divise le nombre X par B, et on conserve le rest R.

On repète le prosecuse en considérant le cosient q obtenu.

le nombre X est obtenu en écrivant tout les restes R_i à partir de la fin et de gauche à droite.

Scanned by CamScanner

Exemple:

$57)_{10} \xrightarrow{?} B_2 \rightarrow 111001$

57	2
1	28 | 2
0	14 | 2
0	7 | 2
1	3 | 2
1	1 | 2
1	0

2.4- Système Binaire: $B_2 = \{0;1\}$

- Poids binaire = Bit = 0;1.
- Bit: la plus petite quantité d'information dans un système binaire.

(1)01.11(1) ← LSB: bit le moin significatif.

MSB ↑ bit le plus significatif.

- Décade: groupement de 4 bits: 1010, 1100,
- Octet: groupement de 8 bits.
- Baud: le nombre de bit d'information reçu par second.
- Mot: une succession de bit traitée comme un tout.

3- Comptage des chiffres:

un chiffre est augmenté de 1 lorsque tout de poids inferieur sont saturés (9 pour B_{10} et 1 pour B_2).

exemple:

00	10
01	11
02	12
03	13
04	14
05	15
06	16
07	17
08	18
09	19

Scanned by CamScanner

4- Règles de calcul :

(a) - Addition :

$0 + 0 = 0$

$0 + 1 = 1 + 0 = 1$

$1 + 1 = 0 + 1$ en retenu.

$1 + 1 + 1 = 1 + 1$ en retenu.

(b) - Soustraction :

$0 - 0 = 0$

$1 - 0 = 1$

$0 - 1 = 1 + 1$ en emprunt.

$0 - 1 - 1 = 0 + 1$ en emprunt.

(c) - Multiplication :

$0 \times 0 = 0$

$0 \times 1 = 1 \times 0 = 0$

$1 \times 1 = 1$

Scanned by CamScanner

Chapitre II : Fonctions Logiques Elémentaires.

I_ Variables et fonctions logiques:

1_ Définition:

Variable logique: une grandeur variable qui ne peut prendre qu'un nombre fini état

Variable logique Binaire: ne peut prendre que deux états.

Ex: VRAi (1) / FAUX (0) ; Allume (1) / éteinte (0)

Fonction logique: étant donné une fonction logique $f(n_1 \to n_n)$ est une fonction qui pour chaque combinisaus des variables prend un état binaire 0; 1.

Exemple:

a	b	c	f(a,b,c)
0	0	0	
0	0	1	
0	1	0	
1	0	0	

2^8 fonctions $= 2^{(2^3)}$

2^n combinaisons.

1.2_ fonction uncomplétement définies:

Quand sa valeur est indifferente ou non spécifier pour certaines combinaisons des variables.

2_ Représentation des fonctions logiques:

Les fonctions peuvent être représenter par:

- les tables de vérité.
- des tables de Karnaugh.
- des equations logiques.
- Equivaleuts décimaux.

2.1_ Table de vérité:

$(x_1 \to x_n)$ VL → Variable logique

TV { n+1 colonnes, n VL, n FL } ⇒ 2^n lignes.

Exemples: TV d'une fonction logique où l'équivalent décimal est premier (3 variables Logi.)

	a	b	c	f(a,b,c)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

2.2 Table de Karnaugh:

Voir plus loin.

2.3 Représentation par des équations logq: forme canonique ($\Sigma(\pi)$).

ex:

	a	b	c	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

$f(a,b,c) = \underbrace{\bar{a}\bar{b}\bar{c}}_{\text{min terme}} + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$

- 2nd forme canonique ($\pi(\Sigma)$)

$f(a,b,c) = (a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})(\bar{a}+\bar{b}+c)$

2.4 Représentation par des équivalents décimaux:

Ex. précédent:

$f(a,b,c) = \{0; 2; 4; 7\}$.

3 Simplification des fonctions logiques:

C'est écrire la fonction sous forme d'une

Scanned by CamScanner

expression plus simple.

3.1 1ère méthode : Réduction algébriques

utilise d'algèbre de Bode.

Axiomes :

- Eléments neutres : $A+0=A$.
  $A.1=A$.
- Commutativité : $A+B=B+A$.
  $A.B=B.A$.
- associativité : $A+(B+C)=(A+B)+C=A+B+C$.
  $(A.B)C=A(B.C)=A.B.C$.
- Distributivité : $A.(B+C)=A.B+A.C$. / +.
  $A+(B.C)=(A+B)(A+C)$ + / .
- Indempotence : $A+A+\dots+A=A$.
  $A.A.A\dots.A=A$
- Complément : $\bar{\bar{A}}=A$.
  $A+\bar{A}=1$.
  $A.\bar{A}=0$.

Théorème de Demorgan :

$A+AB=A$.

$A\bar{B}+BA=A$.

$A(A+B)=A$.

$(A+\bar{B})(A+B)=A$.

3.2 Table de Karnaugh :

La géométriques :

méthode :
- construire la T.K.
- rechercher des groupement de 2; 4; 8; ...
- des variables qui changent d'états sont éliminées.

Table de Karnaugh :

Ex : $f(a,b,c)=\{0;2;5;8;9;10;11;13\}$.

Scanned by CamScanner

4 VL → 2 VL en lignes (00; 01; 11; 10)
4 VL → 2 VL en colonnes

00	00
01	01
10	11
11	10

B. neutre

C. B. Reflechi / Gray.

Scanned by CamScanner

4- Opérateurs logiques Fondamentaux:

3 opérateurs logiques fondamentaux:
- et, ou, complément (NON)

4.1- Opérateur inversion: "NON"

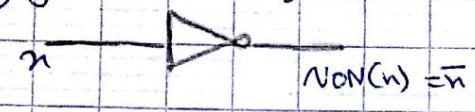
x V.L , $x \in B_2$.

$NON(x) = \bar{x}$

- tableau de Vérité:

x	NON(x)
0	1
1	0

- Logigramme (Symbole).



4.2- Opérateur "Et":

ET: AND; noté "."

$x, y \in B_2 \Rightarrow x.y = x \text{ ET } y = x \text{ AND } y$.

- tableau de Vérité:

x	y	$x.y$
0	0	0
0	1	0
1	0	0
1	1	1

$x \text{ AND } y = x.y$

- Logigramme:

4.3- Opérateur "ou", OR:

$x, y \in B_2$; $x \text{ OU } y = x \text{ OR } y : x + y$.

- tableau de Vérité:

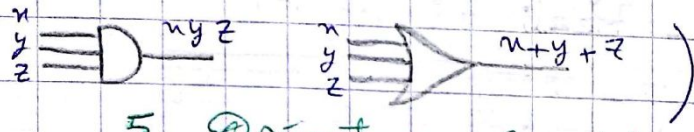
Scanned by CamScanner

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

• Logigramme :

Remarque

"ET" et "OU" peuvent avoir plusieurs entrées



5_ Opérateurs Complets : "NAND" et "NOR"

5.1_ Opérateur complet "NAND"

NAND = NON(ET)

x NAND $y = \overline{x \text{ AND } y}$

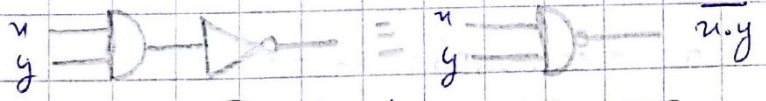
- Notation : NAND.

$x_{NAND} y = \overline{x \cdot y}$

• Table de vérité :

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

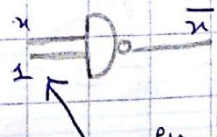
• Logigramme :



5.2_ Réalisations de ET, OU, NON avec NAND

a_ Réalisation de NON.

$\overline{x} = \overline{x \cdot 1}$

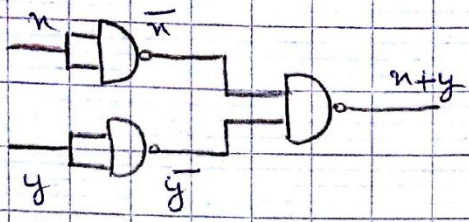


Scanned by CamScanner

$\bar{x} = \overline{x.x}$

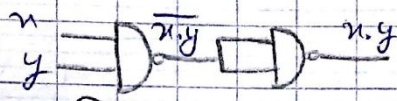
b- Réalisation de OU:

$x+y = \overline{\overline{x+y}} = \overline{\bar{x}.\bar{y}} = \bar{x}$ NAND $\bar{y}$



c- Réalisation de "ET":

$x.y = \overline{\overline{x.y}} = \overline{x \text{ NAND } y}$



5. - Opérateur complet: "NOR"

NOR → NON(OU)

$x, y \in B_2$, x NOR $y = \overline{x+y}$

- Table de Vérité:

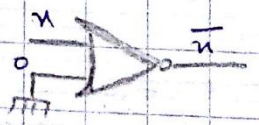
x	y	x+y	$\overline{x+y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

- Logigramme:

a- Réalisation de "NON":

$\bar{x} = \overline{x+x}$

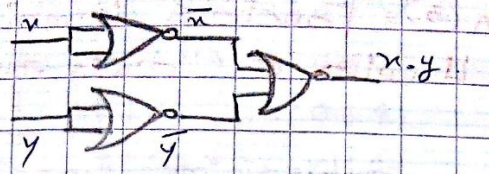
$\bar{x} = \overline{x+0}$



Scanned by CamScanner

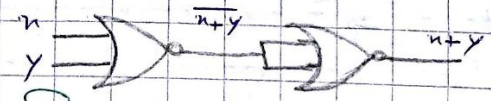
b_ Réalisation de "ET":

$x \cdot y = \overline{\overline{x \cdot y}} = \overline{\bar{x} + \bar{y}}$



c_ Réalisation de "OU":

$x + y = \overline{\overline{x+y}} = \overline{x \text{ NOR } y}$



6_ Opérateurs XOR et XNOR:

6.1 _ opérateur XOR:

XOR : "ou" exclusif

• Table de Vérité:

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

XOR noté $\oplus$.

$x \oplus y = \bar{x}y + x\bar{y}$.

$x \oplus y = 1$ ssi : $x \neq y$.

"XOR" est un opérateur de non équivalence.

• logigramme:

6.2_ Opérateur "XNOR":

XNOR = NON(XOR)

• Table de vérité:

x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

XNOR est noté $\odot$.

$x \odot y = \bar{x}\bar{y} + xy$.

$x \odot y = \overline{x \oplus y}$

$x \odot y = 1$ ssi $x = y$.

Scanned by CamScanner

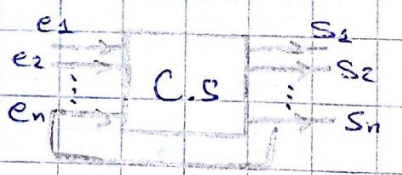
XNOR : opérateur d'équivalence : x, y → xoy

Ch③ : Fonctions Combinatoires Usuelles et Circuits associés

+ Fonctions combinatoires : un circuit est dit combinatoire si ses sorties ne dépend que des combinaisons d'entrée et pas de l'histoire de celles ci → à une combinaisons des variables d'entrée correspond une seule combinaisons de sortie.

+ Le circuit ne conserve aucune mémoire des états précédants.

+ Fonctions (Circuits) séquentielles : un circuit est dit séquentielle si ses sorties dépend des combinaisons d'entrée et l'histoire de celle ci.
→ à une combinaison des variables d'entrée correspond plusieurs combinaisons de sortie.
→ le circuit conserve la mémoire (tout ou une partie) des combinaisons d'entrée et de sortie précédants.



I - Détecteurs d'égalité et comparateurs numériques :

1.1 - Comparateurs de 2 Bits :

x, y 2 bits

- Comparaison de x et y.

Scanned by CamScanner

		=	+	<	>	≤	≥
x	y	f_1	f_2	f_3	f_4	f_5	f_6
0	0	1	0	0	0	1	1
0	1	0	1	1	0	1	0
1	0	0	1	0	1	0	1
1	1	1	0	0	0	1	1

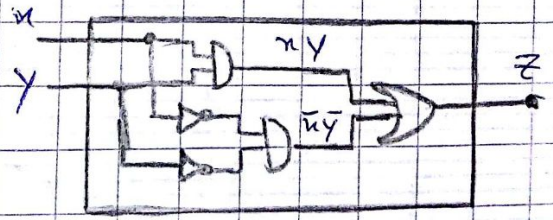
$f_1(x,y) = \bar{x}\bar{y} + xy = x \odot y$.
$f_2(x,y) = \bar{x}y + x\bar{y} = x \oplus y$.
$f_3(x,y) = \bar{x}y$
$f_4(x,y) = x\bar{y}$.
$f_5(x,y) = \overline{f_4(x,y)} = \overline{x\bar{y}} = \bar{x} + y$.
$f_6(x,y) = \overline{f_3(x,y)} = \overline{\bar{x}y} = x + \bar{y}$

1.2 Détecteur d'égalité de 2 Bits:

Réaliser un circuit (CLC) qui détecte l'égalité de 2 Bits?

$z = f(x,y) = \bar{x}\bar{y} + xy = x \odot y$.

x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

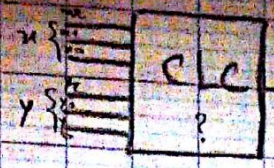


1.3 Détecteurs d'égalité de 2 information codée chacune sur n bits.

Exemple: $n = 3$.

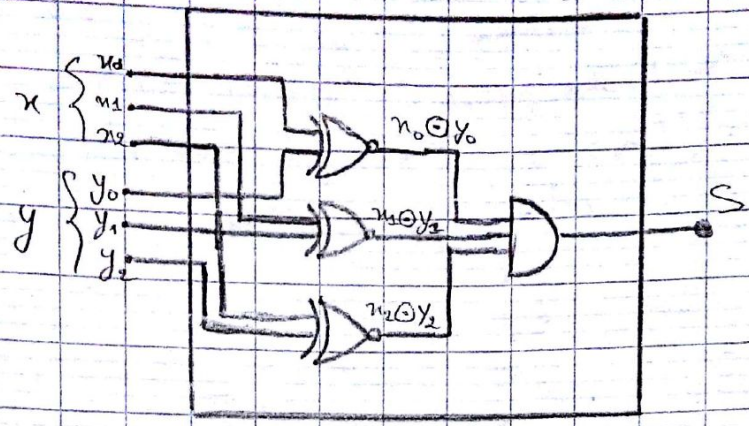
$x = x_2 x_1 x_0$; $y = y_2 y_1 y_0$

Scanned by CamScanner



$S=1$ ssi $\Leftrightarrow (x_0=y_0)$ et $(x_1=y_1)$ et $(x_2=y_2)$.

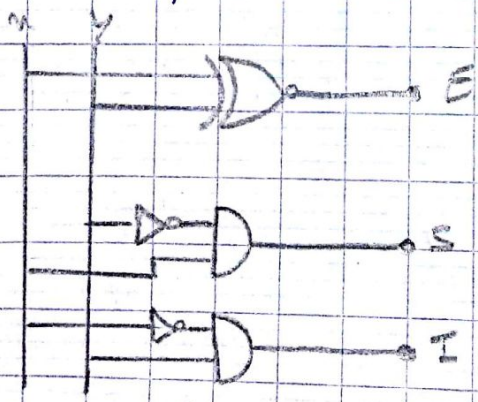
$\Leftrightarrow S = (x_0 \odot y_0).(x_1 \odot y_1).(x_2 \odot y_2)$



1.4- Comparaisons de 2 Bits:

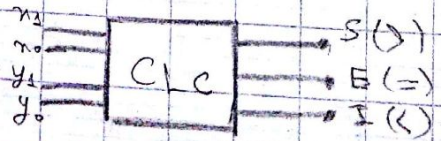
x, y 2 Bits

- $E = x \odot y = \bar{x}\bar{y} + xy$
- $S = x\bar{y}$
- $I = \bar{x}y$



1.5- Comparaison de 2 informations de 2 Bits:

$x = x_1 x_0$; $y = y_1 y_0$.



Scanned by CamScanner

- $E = 1 \Leftrightarrow (x_0 = y_0)$ et $(x_1 = y_1)$

$E = (x_0 \odot y_0) \cdot (x_1 \odot y_1)$

- $S = 1 \Leftrightarrow x > y$

$\Leftrightarrow (x_1 > y_1)$ ou $[(x_1 = y_1)$ et $(x_0 > y_0)]$

$S = x_1 \bar{y}_1 + [(x_1 \odot y_1) \cdot x_0 \bar{y}_0]$

- $I = 1 \Leftrightarrow x < y$

$\Leftrightarrow (x_1 < y_1)$ ou $[(x_1 = y_1)$ et $(x_0 < y_0)]$

$I = \bar{x}_1 y_1 + [(x_1 \odot y_1) \cdot \bar{x}_0 y_0]$

II - Additionneurs numériques:

2.1 - Demi-additionneur:

Déf: est un circuit qui réalise l'addition de 2 Bits x et y. Le résultat de l'addition de 2 Bits est codé sur 2 Bits

x, y 2 Bits

$x + y = RS$

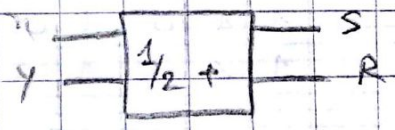
R: Retenus, S: somme

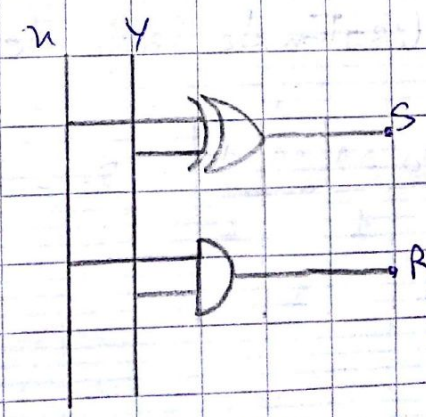
- Table de vérité:

x	y	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = \bar{x}y + x\bar{y} = x \oplus y$

$R = xy$





Circuit d'1/2 add.

Scanned by CamScanner

2.2 Additionneur Complet:

Déf: l'addition simultannée de 3 bits.

Exemple:

$R_2\ R_1\ R_0$ — 1/2 +

$X = X_3 X_2 X_1 X_0$ → 4 bits

$+\ Y = Y_3 Y_2 Y_1 Y_0$ → 4 bits

$R_3 S_3 S_2 S_1 S_0$

add complete

résultat codé sur 5 bits.

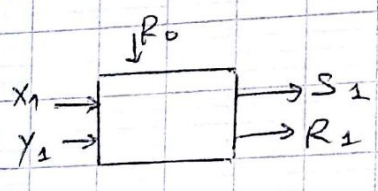
- Additionneurs numériques:
  - demi- Additionneur

x, y 2 bits

$x + y = RS$

↳ Somme LSB

↳ Retenue MSB



Circuit:

X_1	Y_1	R_0	R_1	S_1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$S_1 = \bar{X}_1 \bar{Y}_1 R_0 + \bar{X}_1 Y_1 R_0 + \bar{X}_1 Y_1 \bar{R}_0 + X_1 \bar{Y}_1 \bar{R}_0 + X_1 Y_1 R_0$

$= \bar{X}_1 \cdot (Y_1 \oplus R_0) + X_1 \cdot \overline{Y_1 \oplus R_0}$

$S_1 = X_1 \oplus Y_1 \oplus R_0$

$R_1 = \bar{X}_1 Y_1 R_0 + X_1 \bar{Y}_1 R_0 + X_1 Y_1 \bar{R}_0 + X_1 Y_1 R_0$

$= Y_1 R_0 (\bar{X}_1 + X_1) + X_1 \bar{Y}_1 R_0 + X_1 Y_1 \bar{R}_0$

Utilisation de table de Karnaugh:

S_1:

R_0 \ X_1Y_1	00	01	11	10
0		1		1
1	1		1	

S_1 = même expression que celle de la T.V

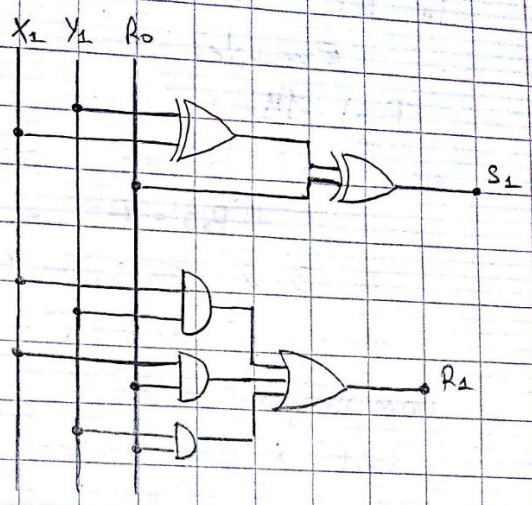
aucun groupement.

Scanned by CamScanner

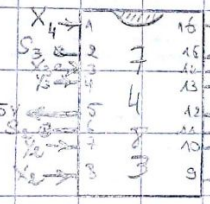
R_1 :

R_0 \ X_1Y_1	00	01	11	10
0			1	
1		1	1	1

$R_1 = X_1Y_1 + X_1R_0 + Y_1R_0$



2.3 Additionneur parallèle de 4 Bits : Circuit intégré 7483



CI 7483

16 Broches

$X = X_4X_3X_2X_1X_0$

$Y = Y_4Y_3Y_2Y_1Y_0$

$R_4S_4S_3S_2S_1$

R_0, 5V, 0V

2.4 Additionneur // de 8 Bits :

$X = X_8 _ X_1$

$Y = Y_8 _ Y_1$

Q : calculer X+Y en utilisant le CI 7483.

$X = X_8 _ X_5 \quad X_4 _ X_1$ ← CI 7483

$+ \; Y = Y_8 _ Y_5 \quad Y_4 _ Y_1$

CI 7483 — $R_8S_8S_7S_6S_5 \quad R_4S_4S_3S_2S_1$

Scanned by CamScanner

III- Soustracteurs numériques:

3.1- Représentation des nombres signés en base 2

* le signe d'un nombre en base 2 est défini par le bit le + fort.

Exemple:

(BS) 1101

Bit de signe

BS = 0 → positif.

BS = 1 → négatif.

3.2- Equivalent binaire d'un nombre décimal négatif:

Complément à 2:

Soit x un nombre positif

L'équivalent binaire de $-x)_{10}$ est le complément à 2 de x.

déf:

le complément à 2 d'un nombre binaire est le resultat obtenu quand chaque bit du nombre est inversé et lorsqu'on ajoute 1 au bit le plus faible.

Exemple:

$6)_{10} = 0110$

$-6)_{10} = 1001 + 1 \Rightarrow 1010$

$-X \rightarrow X$: on retranche 1 et on inverse.

3.3- Soustracteur numérique:

Soient $X, Y > 0$;

calculer $X-Y$?

$X = X_3 X_2 X_1$

$Y = Y_3 Y_2 Y_1$

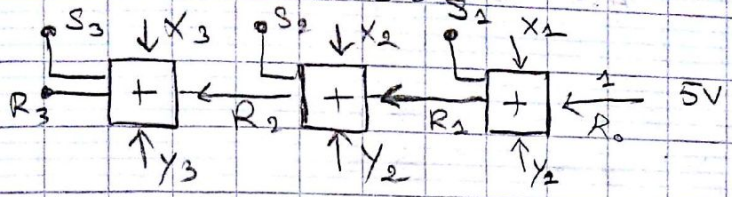
$X-Y = X + (-Y)$

$X-Y = X_3 X_2 X_1 + \bar{Y}_3 \bar{Y}_2 \bar{Y}_1 + 001$

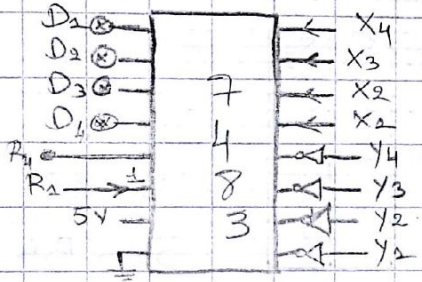
Scanned by CamScanner

$$
\begin{array}{r}
+\ X_3\ X_2 X_1 \\
+\ \overline{Y}_3\ \overline{Y}_2\ \overline{Y}_1 \\
\hline
\quad 0\ 0\ 1
\end{array}
$$

Schéma Bloc:



3.4_ CI7483 branché en Soustracteur:



Exercice:

Comment utiliser le CI7483 en additionneur et en soustracteur?

Solution:

En entrée on utilise une entrée de commande E.

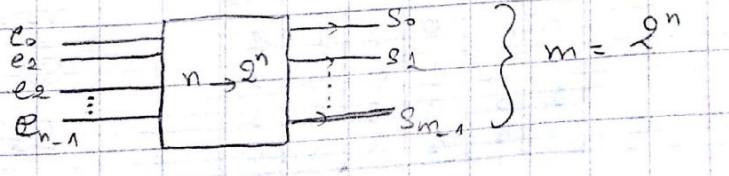
E=0 → additionneur

E=1 → soustracteur)?

IV_ Décodeurs binaires

4.1_ Définitions:

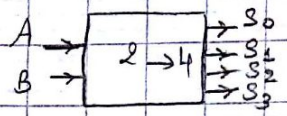
un décodeur binaire est un circuit combinatoire à n entrée (entrée d'adresse) et 2^n sortie ou une seule sortie qui t'active à la fois, un décodeur identifié un code particulier.



Scanned by CamScanner

les entrées sont écrites en binaire naturel, le rang de la sortie active correspond à la valeur binaire de la combinaison affichée sur les entrées d'adresse.

4.2 Décodeur 2 → 4 :



T.V :

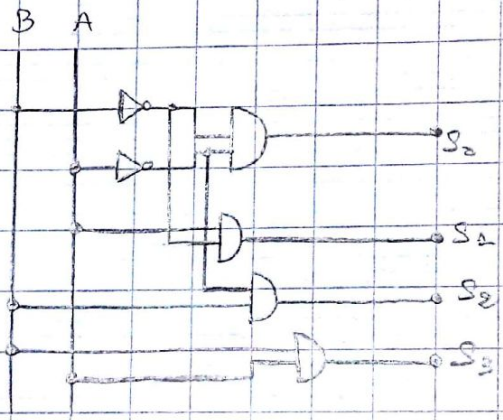
B	A	S_0	S_1	S_2	S_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$S_0 = \bar{B}\bar{A}$

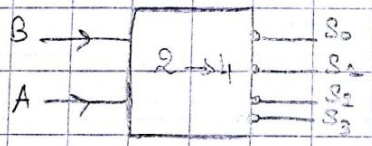
$S_1 = \bar{B}A$

$S_2 = B\bar{A}$

$S_3 = BA$.



4.3 Décodeur 2 → 4 avec sorties actives sur "0".



T.V :

B	A	S_0	S_1	S_2	S_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

$\overline{S_0} = \bar{B}\bar{A} \Rightarrow S_0 = B + A$.

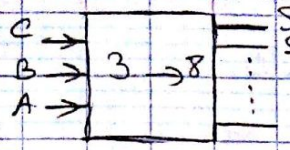
$\overline{S_1} = \bar{B}A$

$\overline{S_2} = B\bar{A}$

$\overline{S_3} = BA$

Scanned by CamScanner

Exemple : Décodeur 3 → 8, sorties actives sur "1"



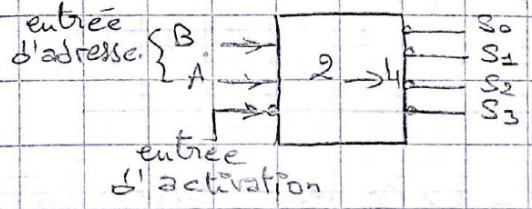
C MSB
A LSB

compléter par exemple

4.4 - Décodeurs avec sortie d'activation

Exemple : Décodeur 2 → 4 avec entrée d'activation G.

Sorties S_0, S_1, S_2 et S_3 actives sur "0"



Fonctionnement :

- G = 0 ⇒ état de BA, le fonctionnement est un décodeur.
- G = 1 ⇒ décodeur inhibé ∀ B, A (inactif).

T.V :

G	B	A	S_0	S_1	S_2	S_3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
	0	1	1	0	1	1
	1	0	1	1	0	1
	1	1	1	1	1	0

$\bar{S}_0 = \bar{G}\bar{B}\bar{A}$

$\bar{S}_1 = \bar{G}\bar{B}A$

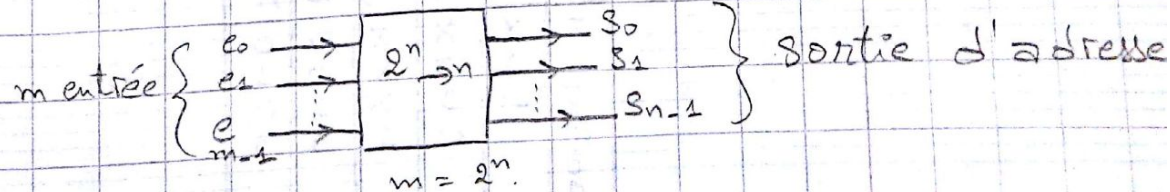
$\bar{S}_2 = \bar{G}B\bar{A}$

$\bar{S}_3 = \bar{G}BA$

V - Encodeur Binaire (codeur) :

5.1 - déf :

Un encodeur binaire c'est l'inverse d'un décodeur.

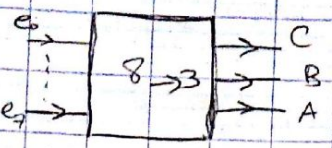


Scanned by CamScanner

→ une seule entrée active à la fois

Exemple:

Codeur 8 → 3.



e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	C	B	A
1								0	0	0
	1							0	0	1
		1						0	1	0
			1					0	1	1
				1				1	0	0
					1			1	0	1
						1		1	1	0
							1	1	1	1

5.2- encodeur prioritaire:

Déf:

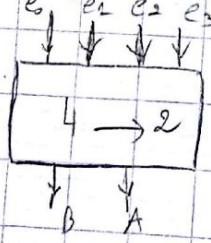
Un encodeur prioritaire est un circuit à 2^n entrée d'informations et n sorties d'adresse, les entrées sont affectées d'une priorité. Les sorties donnent le code binaire du rang de l'entrée active la plus prioritaire.

exemple:

Synthèse d'un encodeur prioritaire 4→2

P. Priorité

$P(e_0) > P(e_1) > P(e_2) > P(e_3)$.



T.V:

	e_0	e_1	e_2	e_3	B	A
8 →	1	X	X	X	0	0
4 →	0	1	X	X	0	1
2 →	0	0	1	X	1	0
1 →	0	0	0	1	1	1

Scanned by CamScanner

6.2 Transcodeurs:

Déf:

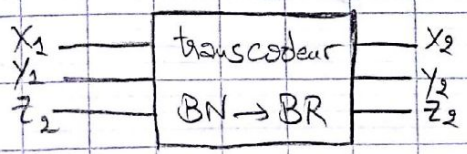
Un transcodeur est un circuit qui permet de passer d'un code à un autre.

a- transcodeur Binaire Naturels → Code de Gray.

Codage sur 3 bits.

- Code binaire naturel. X_1, Y_1 et Z_1.

Code de Gray : X_2, Y_2 et Z_2.



X_1	Y_1	Z_1	X_2	Y_2	Z_2
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

X_1 \ Y_2Z_2	00	01	11	10
0	0	0	0	0
1	1	1	1	1

↘ $X_2 = X_1$

X_1 \ Y_2Z_2	00	01	11	10
0	0	0	1	1
1	1	1	0	0

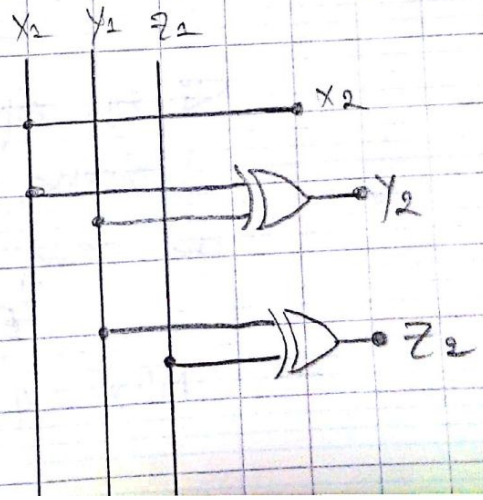
↘ $Y_2 = \overline{X_1}Y_1 + X_1\overline{Y_1}$

$Y_2 = X_1 \oplus Y_1$

Y_1 \ Y_2Z_2	00	01	11	10
0		1		1
1		1		1

↘ $Z_2 = \overline{Y_1}Z_1 + Y_1\overline{Z_1}$

$Z_2 = Y_1 \oplus Z_1$



Scanned by CamScanner

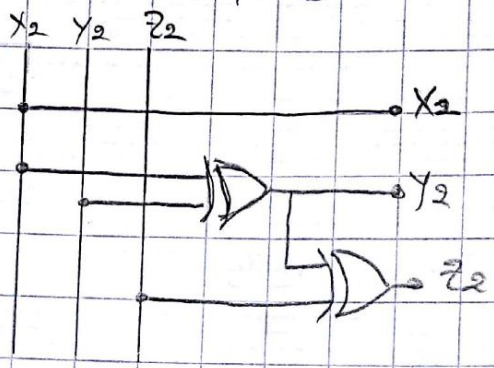
b_ Transcodeur CBG → CBN.

même chose que le transcodeur CBN → CBG.

$X_1 = X_2$.

$Y_1 = X_2 \oplus Y_2$.

$Z_1 = X_2 \oplus Y_2 \oplus Z_2$

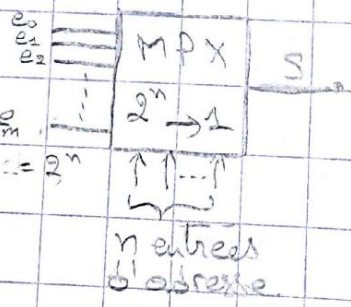


3.2_ Multiplexeurs logiques:

a_ Déf:

Un multiplexeur MPX est un circuit ayant,

- il a 2^n entrées d'information.
- n entrées d'adresse.
- 1 seule sortie.



Si on applique une adresse i sur les entrées d'adresse, l'état de l'entrée de l'information de rang i est affichée en sortie.

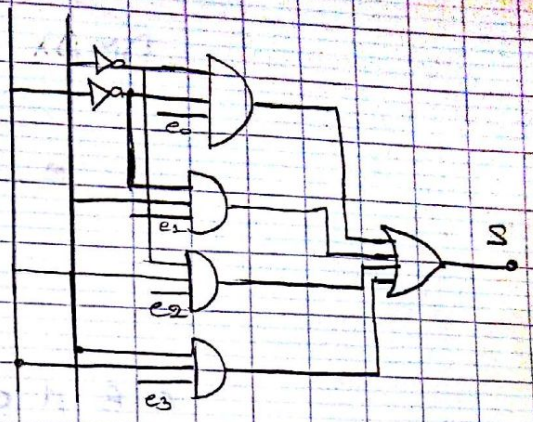
Exemples:

MPX 4 → 1

Scanned by CamScanner

e_0	e_1	e_2	e_3	B	A	S
				0	0	e_0
				0	1	e_1
				1	0	e_2
				1	1	e_3

$S = e_0\bar{B}\bar{A} + e_1\bar{B}A + e_2B\bar{A} + e_3BA$

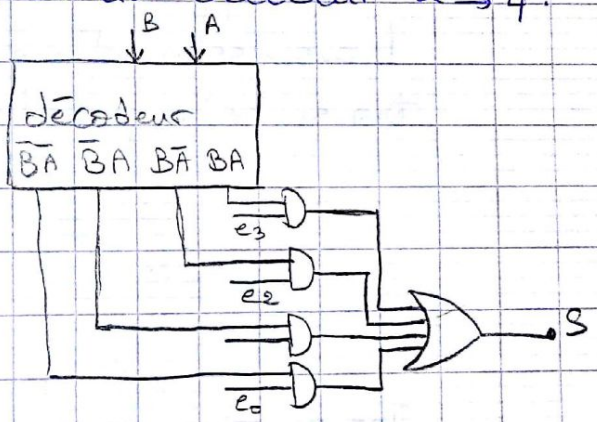


Remarque:

Un MUX permet d'éq. er une information parmis 2^n informations en sortie.

- Autre réalisation:

utilisation d'un décodeur 2 → 4.

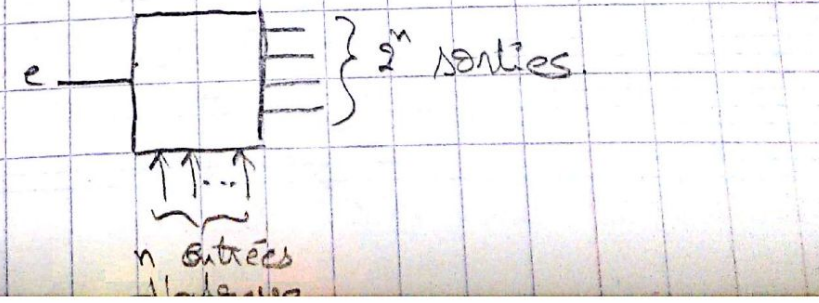


8- Demultiplexeurs:

a- Déf:

Le Demultiplexeurs DMPX réalise d'un fn inverse des multiplexeurs.

Le circuit comporte une entré d'information n entrées d'adresse et 2^n sorties.

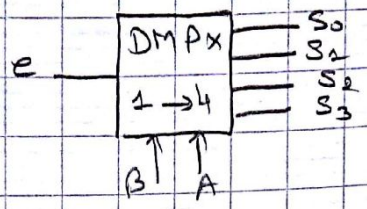


Scanned by CamScanner

l'entrée d'information est équivée sur la sortie dans le rang est affecté sur les adresses.

Exemple:

DMPX 1 → 4.



e	B	A	S_0	S_1	S_2	S_3
	0	0	e			
	0	1		e		
	1	0			e	
	1	1				e

$S_0 = \bar{B}\bar{A}e$

$S_1 = \bar{B}Ae$

$S_2 = B\bar{A}e$

$S_3 = BAe$

b- Application:

transmission d'une information.

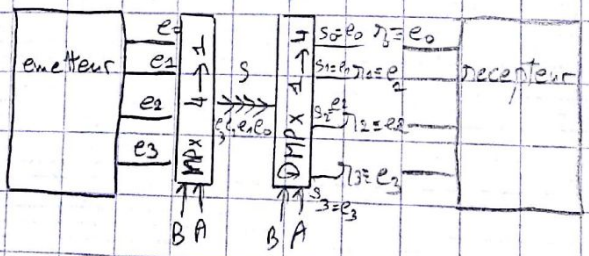
- transmission parallèle.

4 fils pour transmettre e0, e1, e2 et e3

transmission couteuse:

- transmission série

↳ utilisation d'un seul fils de transmission.



si on a 8 → on met 3 entrée d'adresse C B A